

C. AMENDMENTS TO THE CLAIMS

1. (Currently amended) A method for managing allocation of processors in a non-symmetric multiprocessor system for fulfilling at least one request of one or more threads of an application program, the application program comprising a plurality of interacting threads, the multiprocessor system comprising a plurality one or more of general-purpose processors and a plurality one or more of special-purpose processors, each of the special-purpose processors having access to a corresponding local program store, the corresponding local program store being loaded with one or more specific programs, the method comprising the steps of:
 - a. receiving an allocation request from the at least one thread for at least one of the one or more special purpose processors with the corresponding local program store having at least one of the one or more specific programs, wherein the at least one of the one or more specific programs is a requested program and the at least one thread is a requesting thread; ~~receiving- compiling an application program in response to a request for execution of the application program, the application program comprising a plurality of interacting threads, each of the plurality of threads being capable of independently executing an application segment;~~
 - b. granting control of the at least one of the one or more special-purpose processors to the requesting thread; ~~scheduling the plurality of threads on various general-purpose processors and special-purpose processors based on the availability of the processors and the type of request; and~~
 - c. receiving a request for executing the requested program on the at least one of the one or more special-purpose processors from the requesting thread; ~~managing the local program stores of each of the special-purpose processors for complying with processing load, the processing load being dependent on the requests for specific programs and the frequency of such requests.~~

- d. running the requested program on the at least one of the one or more special-purpose processors;
- e. receiving a request to relinquish control of the at least one of the one or more special-purpose processors from the requesting thread; and
- f. releasing the at least one of the one or more special-purpose processors from the requesting thread.

2. (Cancelled)

3. (Cancelled)

4. (Cancelled)

5. (Currently amended) The method as recited in claim 3 1 wherein the step of granting control of the at least one of the one or more special-purpose processors allocating a free special-purpose processor comprises the steps of:

- ~~a. receiving an allocation request from a thread for a processor with a specific program loaded on its local program store;~~
- a. b. searching for the at least one a free of the one or more special-purpose processors with the requested program already loaded on the its corresponding local program store, wherein the at least one of the one or more special-purpose processors is free;
- b. e. allocating the free the at least one of the one or more special-purpose processors with the requested program already loaded on it's the program store to the requesting thread, if such a processor is available; and
- c. d. loading the requested program on the corresponding local program store of at least one of the one or more other a free special-purpose processors and allocating the at least one of the one or more other special-purpose processors to the requesting thread, wherein the one or more other special-purpose

processors are free. ~~if no free special-purpose processor is available with the requested program already loaded on it~~

6. (Cancelled)

7. (Currently amended) A method for allocating one or more special-purpose processors in a multiprocessor computer system running an application program, the application program comprising a plurality of one or more threads, each of the one or more special-purpose processors having access to a corresponding local program store, the corresponding local program stores comprising one or more specific programs, the at least one thread of the one or more threads requesting access to at least one specific program of the one or more specific special programs, ~~the special programs having been stored on the local program stores of the special-purpose processors~~, the method comprising the steps of:

- a. receiving an allocation request from a requesting thread for at least one of the one or more a special-purpose processors ~~with a special-requested program loaded on its local program store~~; wherein the requesting thread being the at least one thread of the one or more threads and a requested program being the at least one specific program of the one or more specific programs;
- b. ~~allocating a special-purpose processor with the requested program loaded on its local program store to the requesting thread, if a free special-purpose processor is available;~~
- b. c. stalling the requesting thread and adding it the requesting thread to a request-queue when [[,]] if no the one or more free special-purpose processors are not available;
- d. ~~checking the request queue for any pending requests, once a special-purpose processor is released by the requesting thread;~~
- c. e. allocating the at least one of the one or more free special-purpose processors to the first requesting thread in the request-queue; when the at least one of the

one or more special-purpose processors is free that requests for a program already loaded on the processor;

d. running the requested program on the at least one of the one or more special-purpose processors; and

f. allocating the free special-purpose processor to the first thread in the request-queue, if none of the threads in the request-queue request for a program already loaded on the processor; and

e. g. receiving relinquishing the control of the at least one of the one or more allocated special-purpose processors from the requesting thread when [[,]] once the at least one of the one or more special-purpose processors becomes idle.

8. (Currently amended) The method as recited in claim 7 wherein the step of allocating a the at least one of the one or more special-purpose processors with the requested program loaded on its local program store to the requesting thread [[,]] if a free special-purpose processor is available, comprises the steps of:

a. searching for a the at least one of the one or more free special-purpose processors with the requested program already loaded on its corresponding local program store, when the at least one of the one or more special-purpose processors is free;

b. allocating the free at least one of the one or more special-purpose processors with the requested program already loaded on its local program store to the requesting thread, if such a processor is available; and

c. loading the requested program on the corresponding local program store of a at least one of the one or more other free special-purpose processors and allocating it the at least one of the one or more other special-purpose processors to the requesting thread [[,]] wherein the one or more other special-purpose processors are free if no free special-purpose processor is available with the requested program already loaded on its local program store.

9. (Currently amended)The method as recited in claim 8 5 wherein the step of loading the requested program comprises the steps of:

- a. virtually evicting the one or more specific programs on the corresponding local program stores of all the one or more of the other free special-purpose processors until at least one of the a one or more other free special-purpose processors with enough space on it's the corresponding local program store to fit the requested program is identified;
- b. creating the enough space by actually evicting the one or more specific programs on the local program store of the at least one of the one or more other identified free special-purpose processors;
- c. loading the requested program in the enough space created on the at least one of the one or more other free special-purpose processors; and
- d. allocating the at least one of the one or more other free special-purpose processors to the requesting thread.

10. (Currently amended)The method as recited in claim 9 wherein the step of virtually evicting the one or more specific programs from the corresponding local program stores of the one or more other free special-purpose processors is carried out in at least one of least-recently-used order, least-frequently-used order ~~or~~ and first-in-first-out order.

11.(Currently amended) The method as recited in claim 9 wherein the step of virtually evicting the one or more specific programs from the corresponding local stores of the one or more other free special-purpose processors further comprises the use of task information while creating the enough space on the at least one of the one or more free special purpose processors, the task information being information regarding at least one of task priority, task execution time, task pending time and program relevance.

12.(Currently amended)The method as recited in claim 7 8 wherein the step of ~~allocating the special-purpose processor to the first thread in the request queue, if none of the threads in the request queue request for a program that is already loaded on the local program store of the special-purpose processor~~ loading the requested program on the corresponding local program store of at least one of the one or more other free special-purpose processors and allocating the at least one of the other one or more free special-purpose processors to the requesting thread, comprises the steps of:

- a. virtually evicting the one or more programs on the corresponding local program store of the at least one of the one or more other free special-purpose processors to create enough space for fitting in the requested program;
- b. creating the enough space for fitting in the requested program on the at least one of the one or more other free special-purpose processors by actually evicting the one or more specific programs;
- c. loading the requested program in the enough space created on the at least one of the one or more other free special-purpose processors; and
- d. allocating the at least one of the one or more other free special-purpose processors to the requesting thread.

13.(Currently amended)The method as recited in claim 12 wherein the step of virtually evicting the one or more specific programs from the corresponding local stores of the one or more other free special-purpose processors is carried out in at least one of least-recently-used order, least-frequently-used order or with first-in-first-out order.

14.(Currently amended)The method as recited in claim 12 wherein the step of virtually evicting the one or more specific programs from the corresponding local program stores of the one or more other free special-purpose processor further comprises the use of task information while creating the enough space on the at least one of the one or more free special-purpose processors, the task information being information

regarding task priority, task execution time, task pending time and program relevance.

15. (Original) The method as recited in claim 7 wherein one or more of the steps is embodied in a computer program product.
16. (Currently amended) A system for managing allocation of processors in a non-symmetric multiprocessor environment, the non-symmetric multiprocessor environment comprising ~~a plurality of one or more~~ general-purpose processors and a ~~plurality of one or more~~ special-purpose processors, each of the one or more special-purpose processors having access to a corresponding local program store, the system comprising:
- a. a compilation service for compiling an application program in response to a request for execution of the application program, the application program comprising a plurality of ~~interacting~~ threads;
 - b. a processor allocation service for scheduling and synchronizing the plurality of threads on various one or more general-purpose processors and one or more special-purpose processors; and
 - c. a local program store managing service for managing the corresponding local program stores of each of the one or more special-purpose processors for ~~complying with processing load~~.
17. (Cancelled)
18. (Cancelled)
19. (New) The method as recited in claim 1 wherein step of granting control of the at least one of the one or more special-purpose processors having the requested program in the local program store to the requesting thread comprises creating a data structure containing information for identifying the at least one of the one or

more special-purpose processors and the corresponding local program store having the requested program.

20. (New) The method as recited in claim 1 wherein the step of granting control the of the at least one of the one or more special-purpose processors further comprises blocking the requesting thread, the requesting thread being added to a request queue wherein one or more special-purpose processors is not available to complete the allocation request.
21. (New) The method as recited in claim 9 wherein the step of creating the enough space by actually evicting the one or more specific programs on the local program store of the at least one of the one or more other free special-purpose processors comprises the step of deleting only the one or more specific programs situated in the enough space identified for the requested program.